Please amend claims 1 and 3-29 to conform to the following clean versions.

A1

1.   A software system with a two tier arrangement for threads support, comprising:

virtual machine including a threads interface layer that provides a standard threads interface in the virtual machine for parallel execution of a plurality of software tasks which are adapted to the virtual machine;

native threads interface layer that includes at least one native threads support routine used by the standard threads interface for adapting parallel execution of the software tasks on a platform which underlies the virtual machine.

---

3.   The software system of claim 1, wherein the threads interface layer maintains a set of context information for each software task in terms of the virtual machine.

A2

4.   The software system of claim 3, wherein each set of context information includes a value for each of a set of virtual machine registers associated with corresponding software task.

5.   The software system of claim 1, wherein the native threads interface layer maintains a set of context information for each software task in terms of the platform.

6.   The software system of claim 5, wherein each set of context information includes a value for each of a set of processor registers associated with the corresponding software task.

7.   The software system of claim 1, wherein the native

- 2 -

threads support routine enables the threads interface layer to suspend a particular software task.

8. The software system of claim 1, wherein the native threads support routine enables the threads interface layer to resume a particular software task.

9. The software system of claim 1, wherein the native threads support routine enables the threads interface layer to wait for completion of a particular software task.

10. The software system of claim 1, wherein the native threads support routine enables the threads interface layer to yield execution to another software task.

11. The software system of claim 1, wherein the native threads support routine enables the threads interface layer to stop an execution of a particular thread and to clean up a set of structures associated with the particular software task.

12. The software system of claim 1, wherein the native threads support routine enables the threads interface layer to set a priority of a particular software task.

13. The software system of claim 1, wherein the native threads support routine enables the threads interface layer to obtain a priority of a particular software task.

14. The software system of claim 1, wherein the native threads support routine enables the threads interface layer to obtain an identifier of a currently executing software task.

15. The software system of claim 1, wherein the native threads support routine enables the threads interface layer to select a software task for execution.

- 3 -

16. A method for providing threads support for a virtual machine in a software system, comprising the steps of:

providing a threads interface layer in the virtual machine that provides a standard threads interface in the virtual machine for parallel execution of a plurality of software tasks which are adapted to the virtual machine;

providing a native threads interface layer that includes at least one native threads support routine used by the standard threads interface for adapting parallel execution of the software tasks on a platform which underlies the software system.

17. The method of claim 16, wherein the threads interface layer maintains a set of context information for each software task in terms of the virtual machine.

18. The method of claim 17, wherein each set of context information comprises a value for each of a set of virtual machine registers associated with a corresponding software task.

19. The method of claim 16, wherein the native threads interface layer maintains a set of context information for each software task in terms of the platform.

20. The method of claim 19, wherein each set of context information comprises a value for each of a set of processor registers associated with the platform.

21. The method of claim 16, wherein the native threads support routine performs the step of suspending a particular software task.

22. The method of claim 16, wherein the native threads

- 4 -

support routine performs the step of resuming a particular software task.

23.   The method of claim 16, wherein the native threads support routine performs the step of waiting for completion of a particular software task.

24.   The method of claim 16, wherein the native threads support routine performs the step of yielding execution to another software task in response to a request from the threads interface layer.

Cont A2

25.   The method of claim 16, wherein the native threads support routine performs the steps of stopping execution of a particular software task and cleaning up a set of structures associated with the particular software task.

26.   The method of claim 16, wherein the native threads support routine performs the step of setting a priority of a particular software task.

27.   The method of claim 16, wherein the native threads support routine performs the step of obtaining a priority of a particular software task.

28.   The method of claim 16, wherein the native threads support routine performs the step of obtaining an identifier of a currently executing software task.

29.   The method of claim 16, wherein the native threads support routine performs the step of selecting a software task for execution.